Javascript osnove

**Varijabla** je osnovna promjenjiva jedinica u kodu koja moze sadrzati jednostavnu ili kompleksnu vrijednost. Na primjer:  
  
let myVar = 123;  
let myObj = {  
 name: “Branimir”,

phone: “066088929”,

};  
  
**Konstanta** je sustinski vrijednost koja se dalje ne moze promjeniti u toku izvrsavanja programa. Na primjer:  
  
const PI = 3.14;

Obratite paznju da su sve prethodne instrukcije koje dajemo u nasem programu zavrsene sa ;. Javascript moze da radi bez ; na kraju, ali se ovo ipak praktikuje kao dobra praksa zbog polje preglednosti.  
Takodje, obratite paznju na nacin na koji smo pisali nas kod. Dobra praksa je stavljati whitespace prije i posle operatora (npr. znak = u prethodnim slucajevima). Javascript ce ignorisati multi-whitespace, ali mi zelimo da to uvijek bude samo jedan karakter.  
  
**Operatori** su znakovi koje koristimo za osnovne matematicke operacije. Na primjer:  
  
+ sabiranje  
- oduzimanje  
\* mnozenje  
/ dijeljnenje  
% ostatak dijeljenja (modulo)  
  
**Izraz** je logicki zaokruzena cjelina koju masina moze da razumije, a statoji se od varijabli, konstanti i operatora. Elementari primjeri izraza u js:

let result = 5 \* 5;  
  
**Komentari** su dio koda koje se nece izvrsiti prilikom rada. Ovo nam daje mogucnost da ostavimo korisna upozorenja ili objasnjenja nekome ko bude citao nas kod (ili za nas same za buducnost).  
  
U JS komentare mozemo pistati na jednoj liniji:  
  
// ovo je jednolinijski komentar  
  
ili na vise linija:  
  
/\*  
ovo je viselinijski komentar

\*/

**Pravila kod imenovanja varijabli, konstanti i funkcija**

* naziv moze poceti velikim ili malim slovom, dolar znakom ili donjom crtom,
* ostatak mogu sacinjavati slova, brojevi, dolar znak ili donja crta.

Dobra praksa je davati smislena imena varijablama, konstantama i funkcijama.

Primjer loseg imenovanja varijable:  
  
let a = 0.2;

mnogo deskriptivniji (i bolji) naziv:

let vatPercentage = 0.2;  
  
Za imenovanje varijabli i funkcija najcesce se primjenjuje “camel case”, tj. naziv pocinjemo malim slovom, a naredne rijeci su zapocete velikim slovom (prethodni primjer).  
  
Za konstante se najcesce koriste velika slova, a rijeci se razdvajaju donjom crtom. Na primjer:

const VAT\_PERCENTAGE = 0.2;

Javascript vazi za “**case sensitive**” programski jezik, sto znaci da su bitna velika i mala slova - dakle:  
  
vatPercentage i vatpercentage pokazuju na potpuno drugacije varijable.

**Znakovi poredjenja u Javascriptu:**

== jednako  
=== jednako i istog tipa

!= nije jednako

!== nije jednako ili nije istog tipa

> vece od

< manje od

>= vece ili jednako od

<= manje ili jednako od

**Logicki operatori**

&& logicko “i”, znaci da izraz lijevo i izrad desno moraju biti istina kako bi krajnji uslov bio zadovoljen

|| logicko “ili”, dovoljno je da je jedan od izraza tacan

! logicko “ne”, koje vraca suprotno od onog sto je bio originalan rezultat provjere

**Tipovi podataka**  
  
Tipovi su zapravo informacija o onome sto je smjesteno u varijablu. Javascript je po definiciji dinamicki programski jezik, sto znaci da nismo duzni unaprijed odrediti koji tip podataka je smjesten u varijablu, nego to interpreter cita iz same vrijednosti. Ovo dalje znaci da u istu varijablu mozemo pisati razlicite tipove podataka (sto moze biti lose!).  
  
**Lista tipova podataka:**  
  
- string: tekstualni tip, tj. skup karaktera ogradjen znakovima navoda (‘ ili “). Na primjer:  
“John Doe” ili ‘John Doe’  
- broj: cijeli (npr 123) ili decimalni - float (npr 1.23)  
- boolean, odnosno logicki tip, vrijednosti su **true** ili **false**- nizovi, skup podataka u uglastim zagradama. Npr: [1, 2, 3] ili [“prvi”, “drugi”, “treci”]  
- objekti, ovo je tip koji se satoji od kljuc-vrijednost parova, ogradjenih viticastim zagradama. Primjer:

{  
 name: “Branimir”,  
 email: “[branimidragicevic1@gmail.com](mailto:branimidragicevic1@gmail.com)”,  
 age: 37,  
 favouriteBooks: [“Eloquent Javascript”, “You don’t know JavaScript”]

}

- undefined, poseban tip koji kaze da varijabla nema vrijednost - nike definisana. Recimo:  
let iDontHaveValue;  
Varijabla iDontHaveValue zapravo nosi undefined tip.

Tip podataka od neke varijable mozemo dobiti koristeci typeof operator. Na primjer:  
  
let myVar = “Neka vrijednost”;

typeof myVar; // string

**Iskazi uslova (if, else if, else i switch)**

Kada zelimo provjeriti tacnost nekog iskaza, koristimo if petlju.  
  
const LIMIT = 5;  
  
let test = 2;  
if (test < LIMIT) {  
 // kod koji budemo pisali ovdje ce se izvrsiti, posto je uslov zadovoljen  
}

if (test > LIMIT) {  
 // kod koji budemo pisali ovdje ce se nece izvrsiti, posto je uslov nije zadovoljen  
}  
  
Ukoliko imamo neku alternativnu provjeru i logiku koja se treba izvrsiti kada prvi uslov nije zadovoljen, onda koristimo else if:  
  
if (test > LIMIT) {  
 // kod koji budemo pisali ovdje ce se nece izvrsiti, posto je uslov nije zadovoljen  
} else if (test < LIMIT) {

// kod koji budemo pisali ovdje ce se izvrsiti, posto je uslov zadovoljen

}

Konacno, ako imamo slucaj da se neka logika treba uvijek izvristi, ako provjera u if ne bude zadovoljena, onda koristimo else:  
  
if (test > LIMIT) {  
 // kod koji budemo pisali ovdje ce se nece izvrsiti, posto je uslov nije zadovoljen  
} else {  
 // kod u ovom bloku ce se uvijek izvrsiti kada if padne. U else if zavisi opet od tacnosti uslova  
}  
  
Ove provjere mozemo kombinovati, tako da:   
  
if (test > LIMIT) {  
 // kod koji budemo pisali ovdje ce se nece izvrsiti, posto je uslov nije zadovoljen  
} else if (test == LIMIT) {

// kod koji budemo pisali ovdje ce se nece izvrsiti, posto je uslov takodje nije zadovoljen

} else {

// kod u ovom bloku ce se izvrsiti, posto su oba prethodna uslova netacna

}

Switch se koristi slicno kao i if, a primjenuje se kada postoji mnogo uslova za provjeru, pri cemo treba da se izvrsi jedan od njih. Na primjer:  
  
const myTest = “uslov”;  
  
switch(myTest) {  
 case “provjera”:  
 // ukoliko bi myTest sadrzano string “provjera”, ovaj blok bi se izvrsio. U ovom slucaju nece.  
 break;

case “test”:

// ukoliko bi myTest sadrzano string “test”, ovaj blok bi se izvrsio. U ovom slucaju nece.

case “uslov”:

// nas input je upravo string “uslov”, takod da cemo uci u ovaj blok;  
 break;

default:  
 // ako se nista od prethodnog ne izvrsi, uci cemo u ovaj blog. default moze biti izostavljen, tako se nijedan od uslova ne zadovolji.  
}

Obratite paznju na break kljucnu rijec. Ukoliko je ne koristimo, interpreter ce nastaviti da provjerava i ostale uslove i eventualno moze da izvrsi neki. Izostavljanje break kljucne rijeci se ne proporucuje, posto moze izazvati nepredvidivo ponasanje/izvrsavanje koda.

**console.log**

**Console log** koja omogucuje ispis vrijednost u konzolu (bas, command prompt). Izuzetno korisna i cesto se koristi za debug-ovanje (pronalazenje gresaka u kodu/logici).

Zadatak 1:

U nasoj aplikaciji imamo nekoliko mogucnosti za pretplate (subscriptions):

- Basic

- Premium

- Platinum

- Employee

Kada se radi o Employee tipu pretplate, usluga je besplatna ("FREE").

Ostale opcije moraju platiti usluge koriscenja aplikacije putem internet placanja ("CREDIT\_CARD");

Nas zadatak je da napisemo logiku koja ce ispise u konzoli:

"Current subscription type has [ovdje nacin placanja: FREE ili CREDIT\_CARD] payment method"

Zadatak mozete rjesiti na vise (ispravnih) nacina.

Zadatak 2:

Naredni primjer je povezan sa prethodnim. Svaka od pretplata nosi odredjeno snizenje:

- Basic - 20%

- Premium - 40%

- Platinum - 60%

- Employee - 100%

Pretpostavimo da se upravo izvrsava neka narudzba i potrebno je primjeniti popust na ukupan izos prema prethodnoj skali.

Ukupna vrijednost narudzbe:

const totalPrice = 300;

Napisati logiku koja ce ispisati u konzoli:

"Discounted price is: [umanjena cijena za vrijednost popusta]"

Zadatak 1:

const BASIC = “BASIC”;

const PREMIUM = “PREMIUM”;

const PLATINUM = “PLATINUM”;

const EMPLOYEE = “EMPLOYEE”;

const PAYMENT\_FREE = “FREE”;

const PAYMENT\_CREDIT\_CARD = “CREDIT\_CARD”;

const requestWithSubscription = BASIC;  
  
if(requestWithSubscription === EMPLOYEE) {

console.log(`Current subscription type has ${PAYMENT\_FREE} payment method`);

} else {

console.log(`Current subscription type has ${PAYMENT\_CREDIT\_CARD} payment method`);

}

Zadatak 2:

const BASIC = 20;

const PREMIUM = 40;

const PLATINUM = 60;

const EMPLOYEE = 100;

const totalPrice = 300;

let totalPriceWithDiscount = 0;

const requestWithSubscription =”BASIC”;

if(requestWithSubscription === “BASIC”) {

totalPriceWithDiscount = totalPrice - totalPrice \* BASIC / 100;

} else if (requestWithSubscription === “PREMIUM”) {

totalPriceWithDiscount = totalPrice - totalPrice \* PREMIUM / 100;

} else if (requestWithSubscription === “PLATINUM”) {

totalPriceWithDiscount = totalPrice - totalPrice \* PLATINUM / 100;

} else {

totalPriceWithDiscount = 0;

}

console.log(`Discounted price is: ${totalPriceWithDiscount}`);